

# Introduction to



**UNREAL 4 ENGINE**

# What is Unreal Engine 4?

**“Unreal Engine 4 is a suite of integrated tools for game developers to design and build games, simulations, and visualizations.”**

Use for free - 5% Royalty

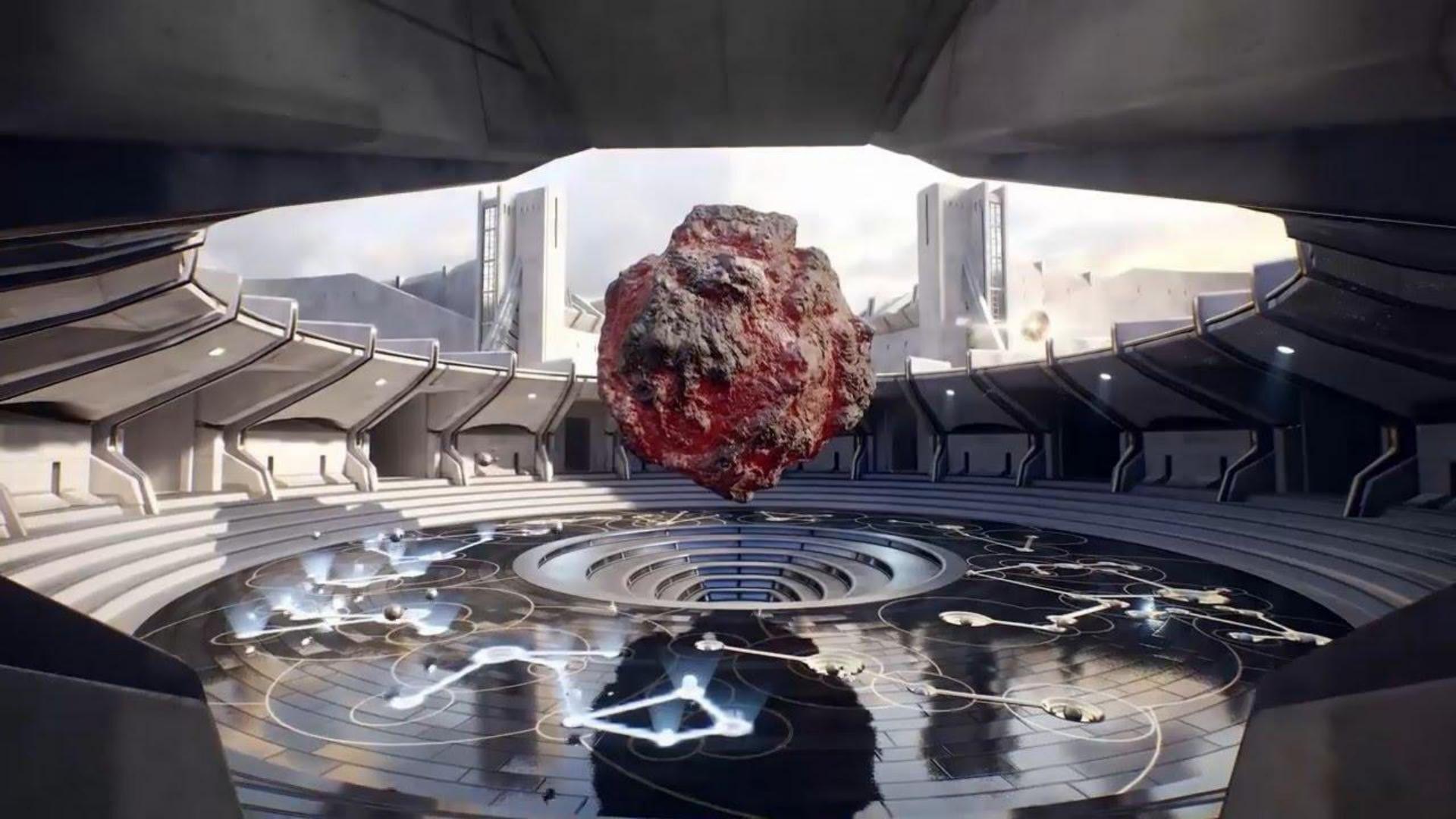
# Stuff Made with UE4





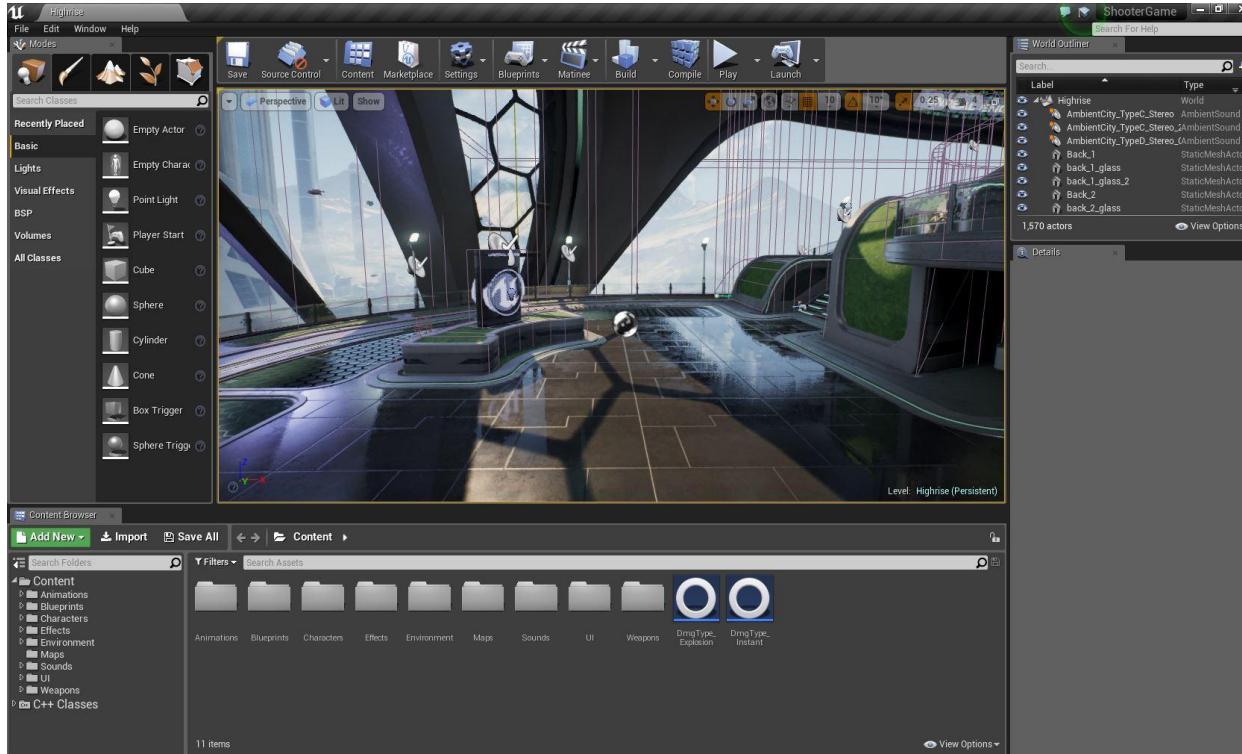




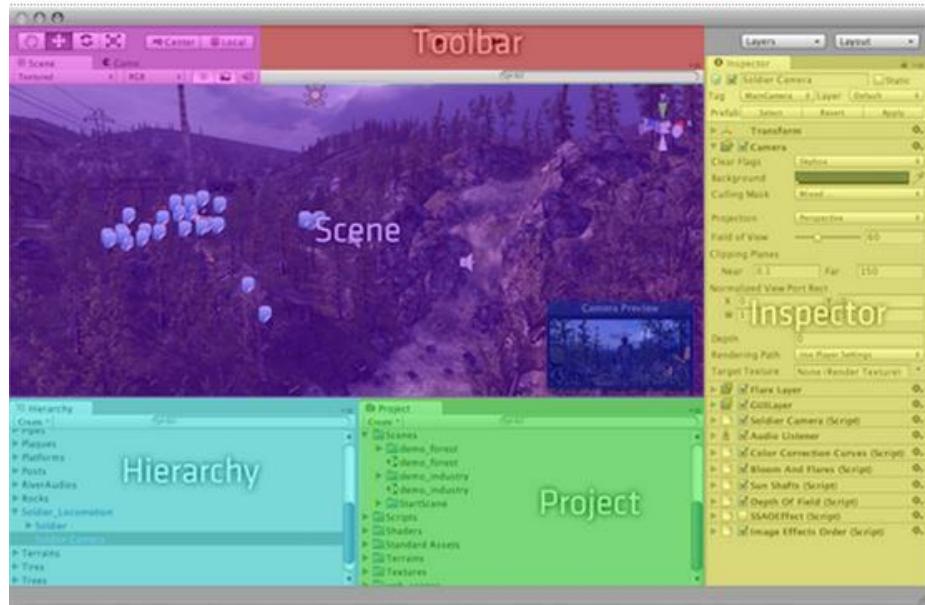




# This is the “Unreal Editor”



# Unity's vs Unreal's Editor



# Wall of Features

- UI Designer
- Skeletal Mesh Animation System
- Sequencer Editor
- Audio Engine
- Behavior Trees
- EQS (Environment Query System)
- Physics Simulation
- Open World Tools
- Landscape + Terrain + Foliage
- Instanced Meshes
- Level Streaming
- Hierarchical Level Of Detail
- Splines
- Matinee + Cinematics
- Media Framework
- Asset Cooker + Packager
- Plugins
- Replay system
- Multiplayer + Networking
- Integration with Steam/XLive/PSN/etc
- Derived/Distributed Data Cache
- “Paper” 2D
- Game Framework
- Blueprints
- Material editor
- Particle systems
- Support for any shading language
- Profiling Tools
- Integration with Visual Studio
- Open Source
- VR/AR support
- **Much much more**

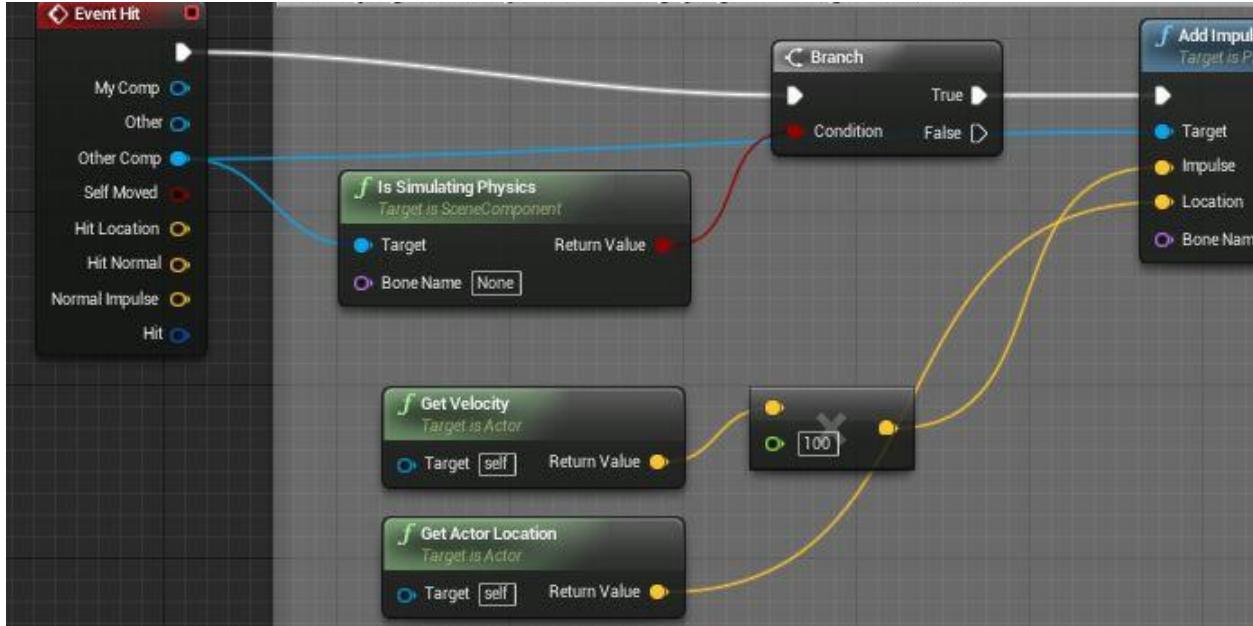
[-] <a href="#">Engine Features</a>	
[+] <a href="#">Rendering and Graphics</a>	
[+] <a href="#">UMG UI Designer</a>	
[+] <a href="#">Skeletal Mesh Animation System</a>	
[+] <a href="#">Sequencer Editor</a>	
[+] <a href="#">Audio and Sound</a>	
[+] <a href="#">Behavior Trees</a>	
[+] <a href="#">Environment Query System</a>	
[+] <a href="#">Physics Simulation</a>	
[+] <a href="#">Open World Tools</a>	
[+] <a href="#">Landscape Outdoor Terrain</a>	
[+] <a href="#">Foliage Instanced Meshes</a>	
[+] <a href="#">Level Streaming</a>	
[+] <a href="#">Hierarchical Level of Detail</a>	
[+] <a href="#">Blueprint Splines</a>	
[+] <a href="#">Matinee and Cinematics</a>	
[+] <a href="#">Media Framework</a>	
[+] <a href="#">Performance and Profiling</a>	
[+] <a href="#">Packaging and Cooking Games</a>	
[+] <a href="#">Replay System</a>	
[+] <a href="#">Built-In Plugins</a>	
<a href="#">Assets and Packages</a>	
<a href="#">Coordinate Space Terminology</a>	
<a href="#">Derived Data Cache</a>	
<a href="#">Directory Structure</a>	
<a href="#">Distributions</a>	
<a href="#">Redirectors</a>	
<a href="#">Taking Screenshots</a>	
[+] <a href="#">Collaboration in Unreal Engine 4</a>	
[+] <a href="#">Paper 2D</a>	
[+] <a href="#">Developer Tools</a>	
<a href="#">Online Beacons</a>	

# What makes it stand out?

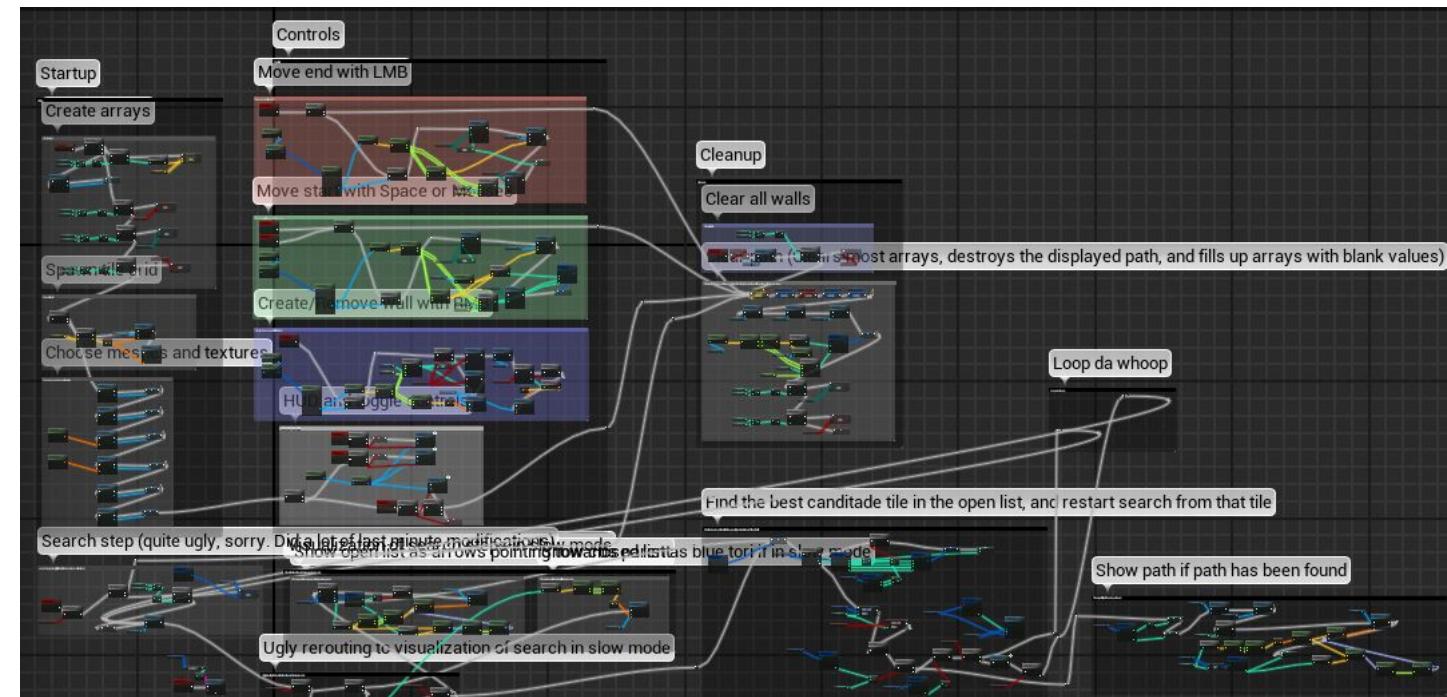
- Blueprints
- Multiplayer
- Shader/Material Editor
- Graphical Fidelity
- Sequencer + Matinee
- Particle Editor
- Derived Data Cache / Build System
- Fast integration with nVidia and other new technologies like Vulkan
- Weekly Livestreams

# Blueprints

Beginner Friendly + Ultra Fast Prototyping



# But they can get pretty chaotic....



# Multiplayer

Replication. It just works, if you dedicate half a day on reading about the topic.

One of the few engine features that is bug-free on the implementation level. The only things that can cause a bug are logical errors.

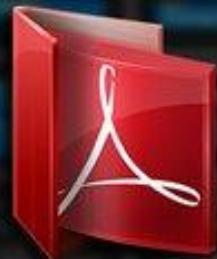


# If you want to get more insight:

**Unreal Engine 4**

**Network Compendium**  
by  
**Cedric 'eXi' Neukirchen**

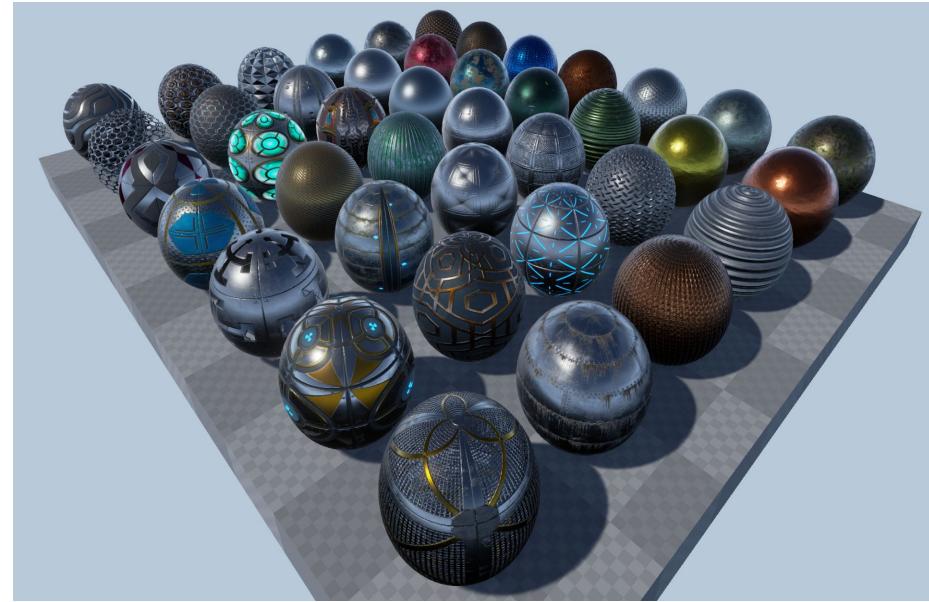
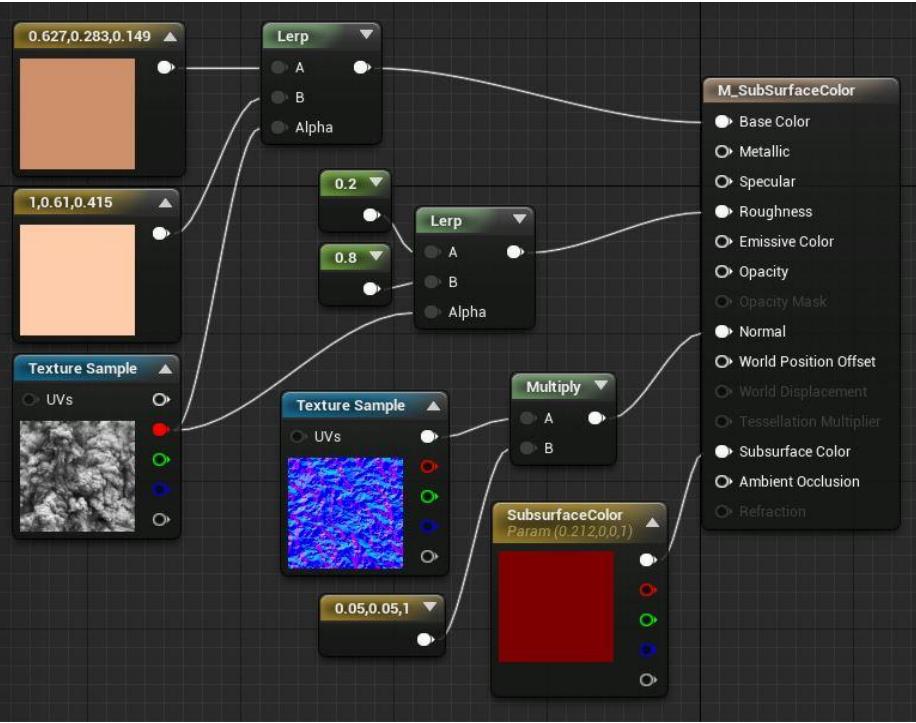
100+ Pages of UE4 Network  
Information to get you started  
with Multiplayer in UE4!



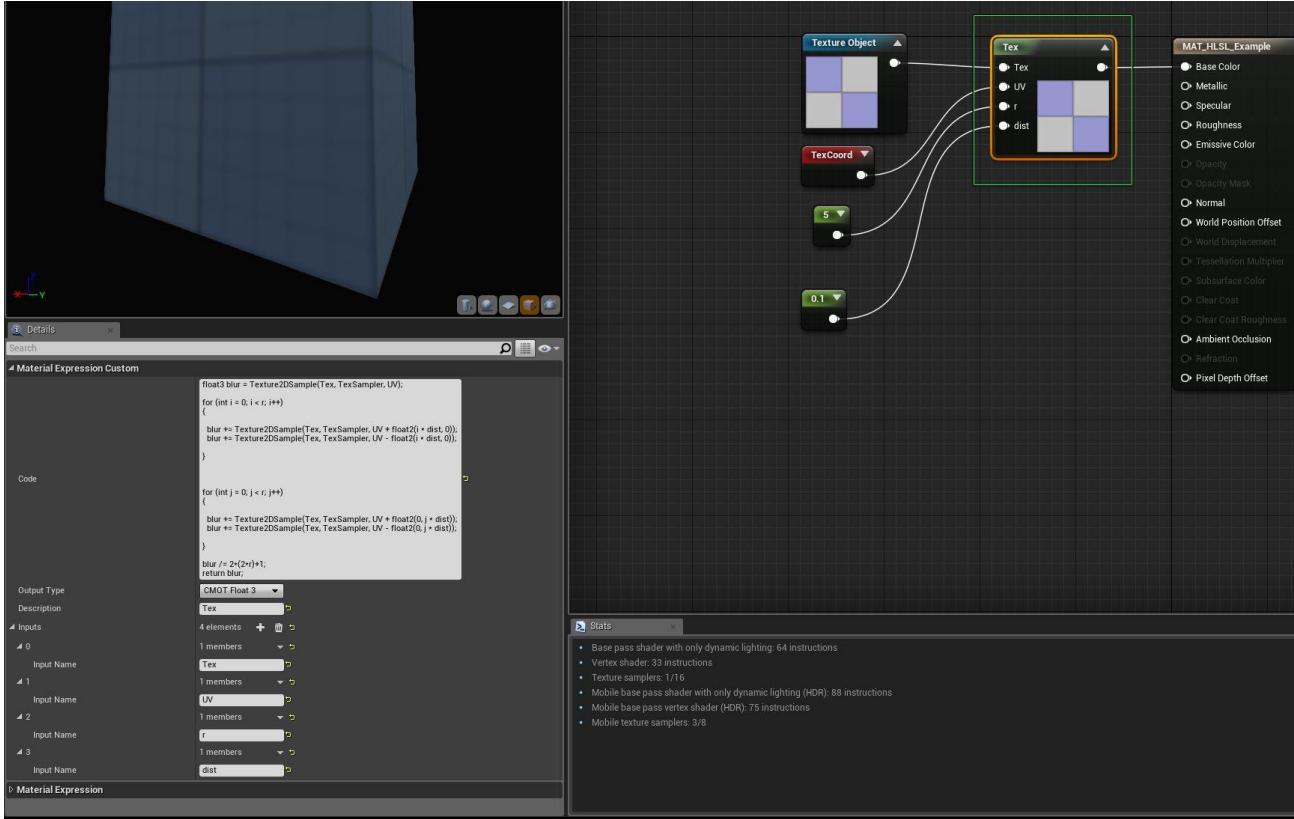
**Download PDF**

- > Information about common Classes
- > Replication, RPCs, Ownership
- > General Subsystem Interface
- > With Blueprint and C++ Examples
- > And much more!

# Material Editor!

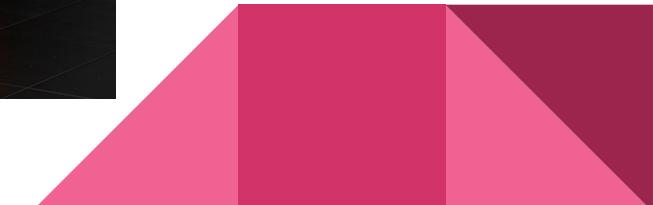


# + Support for custom shaders



# More-than-average Graphical Fidelity

BUT there's a catch: It works only if your 3D people are capable of following specific pipelines.



# Sequencer + Matinee

“Premiere Pro + After Effects” integrated into UE4?



**Modification**

- Brush Edit
- Mesh Paint
- Landscape
- Foliage
- Tex Align

**World Props**

- Transform
- Rotate
- Scale
- Combined
- View
- World Props
- Level Script
- Matinee
- Build
- Compile
- Simulate
- Play

**World**

- 16
- 11.25
- 5
- 4

**Details**

PointLightMovable selected

**PointLight**

Search

**Transform**

Location	X: 1216	Y: 1424	Z: 608
Rotation	X: -0	Y: 0	Z: 0
Scale	X: 1	Y: 1	Z: 1

**Actor**

**Code View**

**Light**

Radius	1024.000
Light Falloff Exponent	8.000
Brightness	3.142
Indirect Lighting Intensity	3.090
Indirect Influence Radius Scale	1.000
Area Light Radius Percentage	0.000
Light Color	R: 255 G: 255 B: 255
Enabled	<input checked="" type="checkbox"/>
Cast Shadows	<input checked="" type="checkbox"/>
Cast Static Shadows	<input checked="" type="checkbox"/>
Cast Dynamic Shadows	<input checked="" type="checkbox"/>
Cast Translucent Shadows	<input type="checkbox"/>
Affect Translucent Lighting	<input checked="" type="checkbox"/>

**Scene Outliner**

Search Actors

Actor Info

- Global PostProcess
- Ground
- Brick
- Knot
- Sphere
- Height Fog
- Player Start
- PointLightMovable
- Sky
- Sun

**Auto**

All Tracks

ElementalDemo > MainShots > Awakening

Search

**DIRECTOR**

**MASTER AUDIO**

OpeningDoor

Whoosh

MoltenFlowSound

Hammer Boost

**Ancient Knight**

**ANIMATION**

WakingFromSleep\_Anim1

RisingFromDust

KnightMarchAnim

KnightBurning

Boosting\_Knight

Transform

Location

Rotation

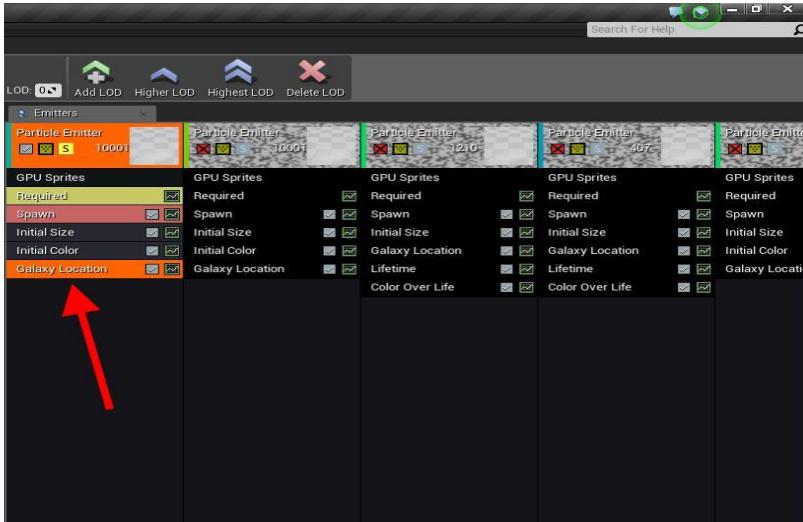
Point Light

Brightness

3.142

# Particle Editor

Many features, event-driven particles, support for external programs e.g. You can import sequences from after effects, or vector fields from Maya.



Find In CB Restart Sim Restart Level Undo Redo Thumbnail Bounds Origin Axis Background Color Regen LOD Regen LOD Lowest LOD Lower LOD Add LOD LOD: 0 Add LOD Higher LOD Highest LOD Delete LOD

ewport

View Time



Emitters

Flames	Flames	Smoke	Embers	Sparks
Required	Required	Required	GPU Sprites	GPU Sprites
Spawn	Spawn	Spawn	Spawn	Spawn
Lifetime	Lifetime	Lifetime	Lifetime	Lifetime
Sphere	Sphere	Initial Location	Initial Size	Initial Size
Initial Size	Initial Size	Initial Size	Sphere	Sphere
Size By Life	Size By Life	Size By Life	Color Over Life	Color Over Life
Color Over Life	Color Over Life	Color Over Life	Conc. Acceleration	Initial Velocity

Details

Search



Color

Color Over Life

Distribution

Constant Curve

Points

0

In Val

DistributionVectorConstantCurve\_0

Out Val

2 elements

+ -

5 members

0.0

X 20.0 Y 5.0 Z 1.0

X 0.0 Y 0.0 Z 0.0

X 0.0 Y 0.0 Z 0.0

Linear

1

In Val

5 members

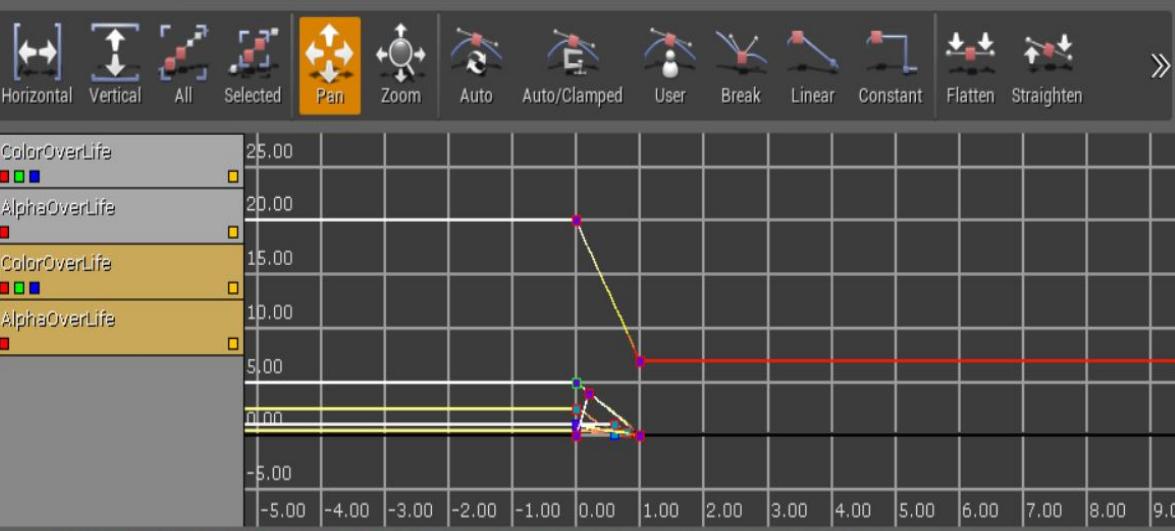
1.0

X 7.0 Y 0.1 Z 0.0

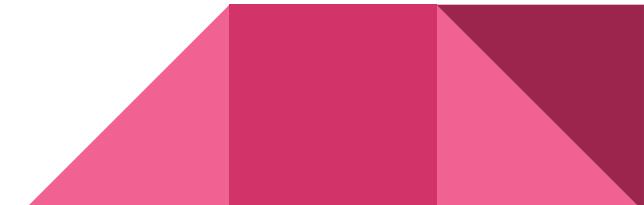
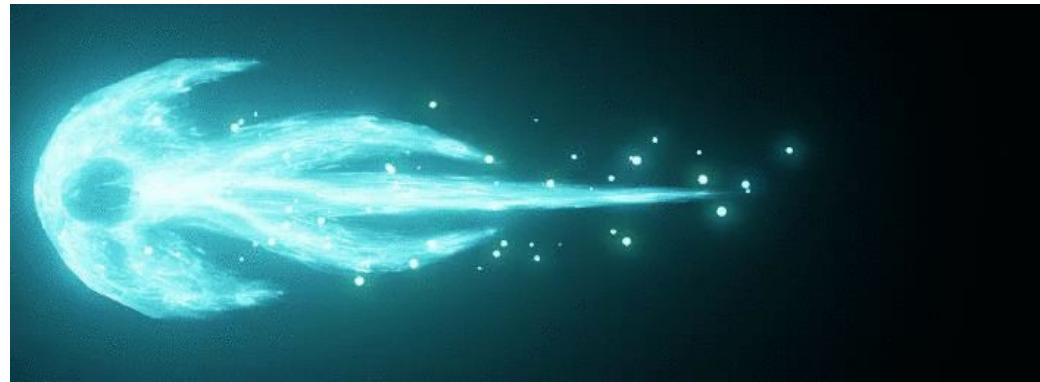
X 0.0 Y 0.0 Z 0.0

X 0.0 Y 0.0 Z 0.0

Curve Editor



# Particles + Materials = Eye candy



# Distributed Workloads and Cache

Basically asset previews, compile cache and other stuff can be stored in some master server with an ssd in order to redistribute fast and avoid the need to re-calculate the same stuff from another machine on the same network.

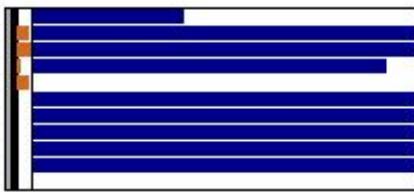
The same technique is used with non-NAS configurations.

File Edit Cache Network Help

Log Swarm Status Settings

Machine

Z3916



Distributed Progress

66.67%

Key

Exporting scene

Lightmass Starting

Emitting Photons

Collecting Photons

Processing Mappings

Exporting

Fast adoption of new technologies



Vulkan.<sup>®</sup>



SteamOS

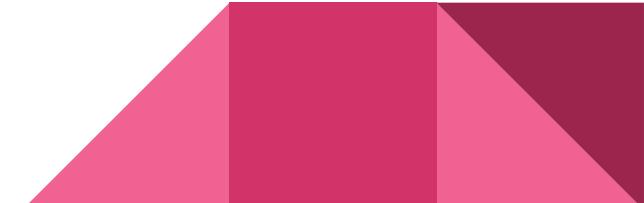


NVIDIA®



# Did you know?

Unreal Engine was the first to adopt Google's Daydream (aka Cardboard v2)!



# Weekly Livestreams

Epic Game's does weekly livestreams, which contain:

- Community Spotlights
- Livestream Announcements
  - New technologies
- The state and future of the engine
- “Guest star” teaching sessions
- Insights on best practises and how Epic does stuff
  - 3-hour or multi-part teaching streams





open source

# Open Source!

You can contribute too! (Like I did)



Flexible C++ subsystem, Shared Pointers + much more data types and design patterns you are going to either love or hate!

Custom nVidia GameWorks integration forks available for free!

# Got a bug that you can't squash?

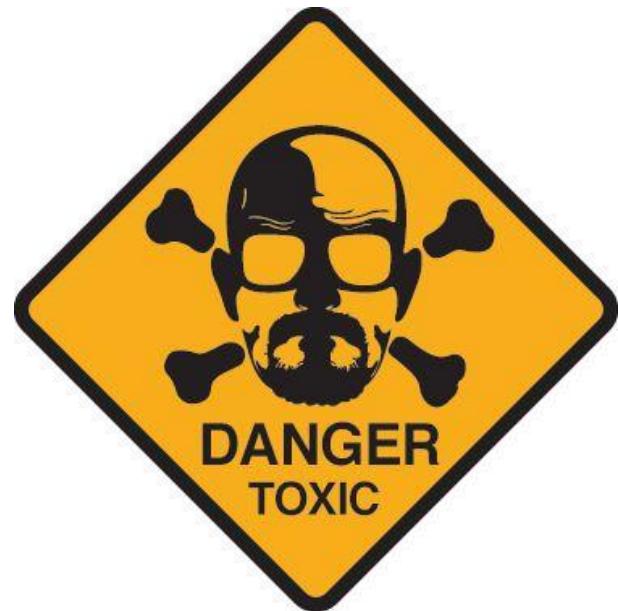
- DON'T BE AFRAID TO LOOK INTO THE ENGINE'S CODE!
- It may be something simple that you can fix and contribute yourself...

...or something way too hard to fix. In that case, you should submit something to the bugtracker.

- There are people like "Coconut Lizard" that use profiling tools and actually boost performance of the engine per engine update!

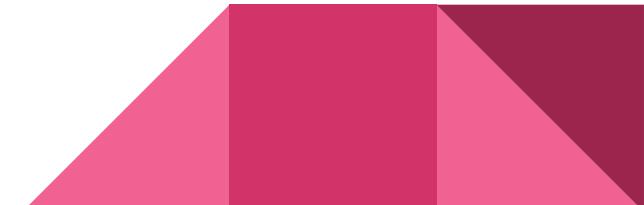
# Bad Parts of Open Source

- Many undocumented places
- Features added > bugs fixed  
(median of every release)
- Contributing guidelines are a bit controversial  
and non-existent :(



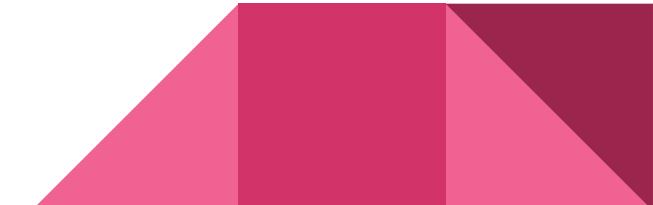
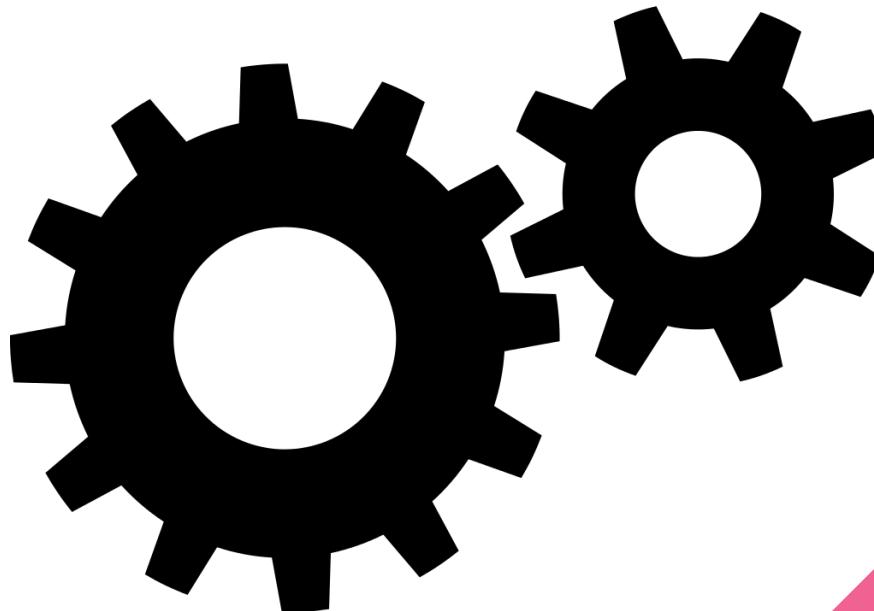
# Forget the C++ you used to know!

- Not your standard C++ project. It's a multi-module beast with it's own build tool and takes much time to compile even with a single header change.
- Many, many, many classes and types of data.
- Learn to use the engine via blueprints first is what I recommend.
- Integration with VS + support for compilation is a big plus!



# Game Framework Walkthrough

...or where do I start if I want to make a game?



# Actor

“An Actor is any object that can be placed into a level. Actors are a generic Class that support 3D transformations such as translation, rotation, and scale.

Actors can be thought of, in one sense, as containers that hold special types of Objects called Components. Different types of Components can be used to control how Actors move, how they are rendered, etc. The other main function of Actors is the replication of properties and function calls across the network during play.”

# GameMode, GameState, GameInstance

*Self-explanatory* but flexible.

They can be used for anything, but in the end, it comes down to good, maintainable and clean software design, **so PLEASE DO NOT VIOLATE.**

# Pawn

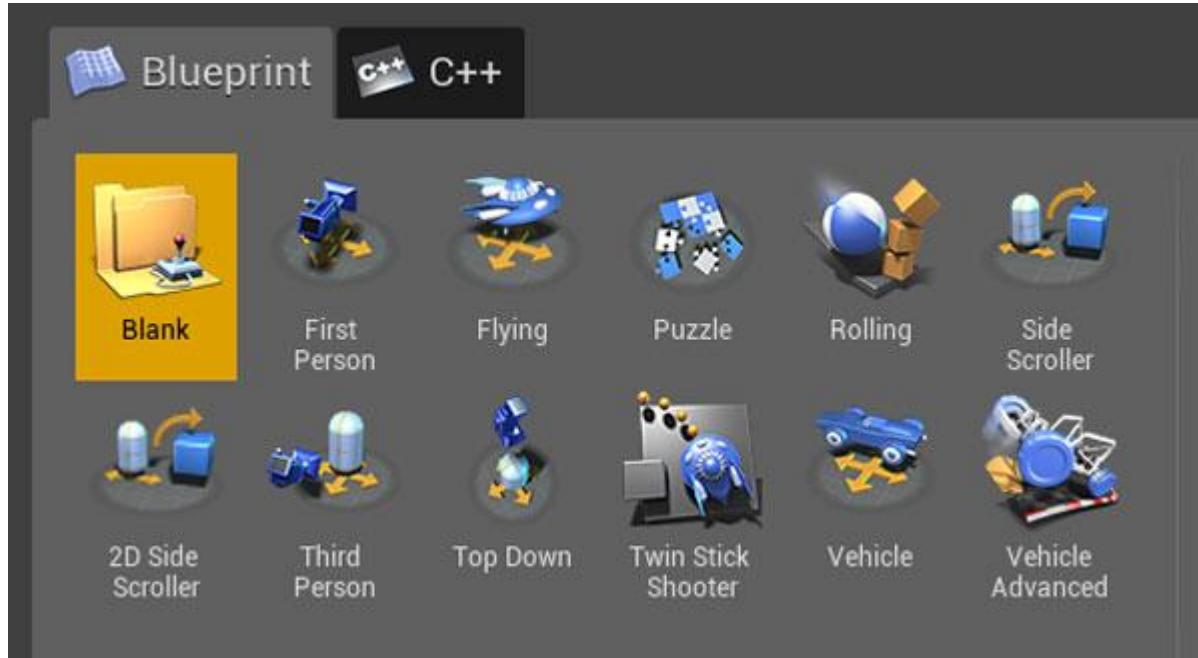
The **Pawn** class is the base class of all Actors that can be controlled by players or AI

Many defaults & templates:



- 1st/3rd person character
  - Sidescroller
  - Top-Down/RTS look
    - Flying
- Candy-Crush like + more!

# Example templates



# Fraction of example projects

## Essentials



Using Content Examples >



Level Design Content Examples >



Geometry Editing Content Examples >



Import Options Content Examples >



Navmesh Content Examples >



Math Hall Content Examples >

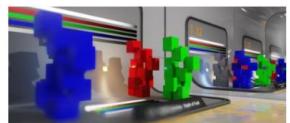
## Rendering



Materials Content Examples >



Lighting Content Examples >



Post Processing Content Examples >

## Animation



Animation Content Examples >



Morph Targets Content Examples >



Effects Content Examples >



Pivot Painter Content Examples >



Cloth Content Examples >



Dynamic Scene Shadows >



Reflections Content Examples >



Decals >

More at:

<https://docs.unrealengine.com/latest/INT/Resources/ContentExamples/>

# Controllers

(= Brains of Pawns)

**Controllers** are non-physical Actors that can possess a Pawn

Controllers receive notifications for many of the events occurring for the Pawn they are controlling. This gives the Controller the opportunity to implement the behavior in response to this event, intercepting the event and superseding the Pawn's default behavior.

By default, there is a one-to-one relationship between Controllers and Pawns.

# Cameras

The engine has a “Camera Manager” which accepts new view targets, in order to change what the player “sees”.

These view targets can be a camera actor, a camera component, inside actor instances, or as “singleton” objects which exist on each level and are accessible by the level blueprints.

# User Interface

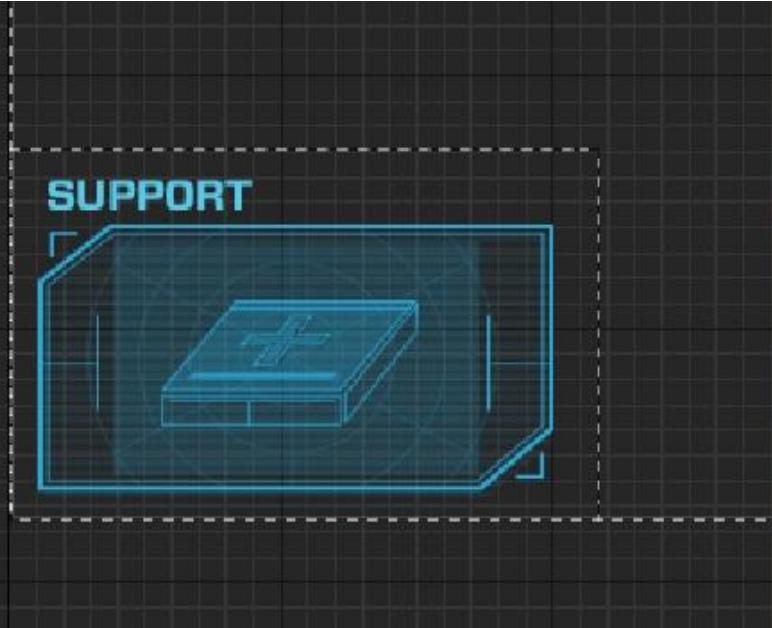
UE4 recently updated it's ecosystem with the addition of UMG (Unreal Motion Graphics), stepping away from Adobe Scaleform, thus providing better integration and bigger flexibility inside the engine.

# UMG

The whole ecosystem is very non-coder friendly and is mostly used with blueprints. There are some interesting plugins that let you programmatically bootstrap the whole interface.

Responsiveness is something that is not well-made (compared to let's say what bootstrap does). You can programmatically calculate new transofrm and scale on construct but it's not a convenient solution.

# 3D Widgets!



# Pretty vs Ugly UMG



# HUD

A non-component or editor based UI tool. Mostly used for debugging and programmatic rendering of stuff like object outlines, crosshairs, text projected from the 3D world to the screen. For example a **minimap**.

\*Generally ugly

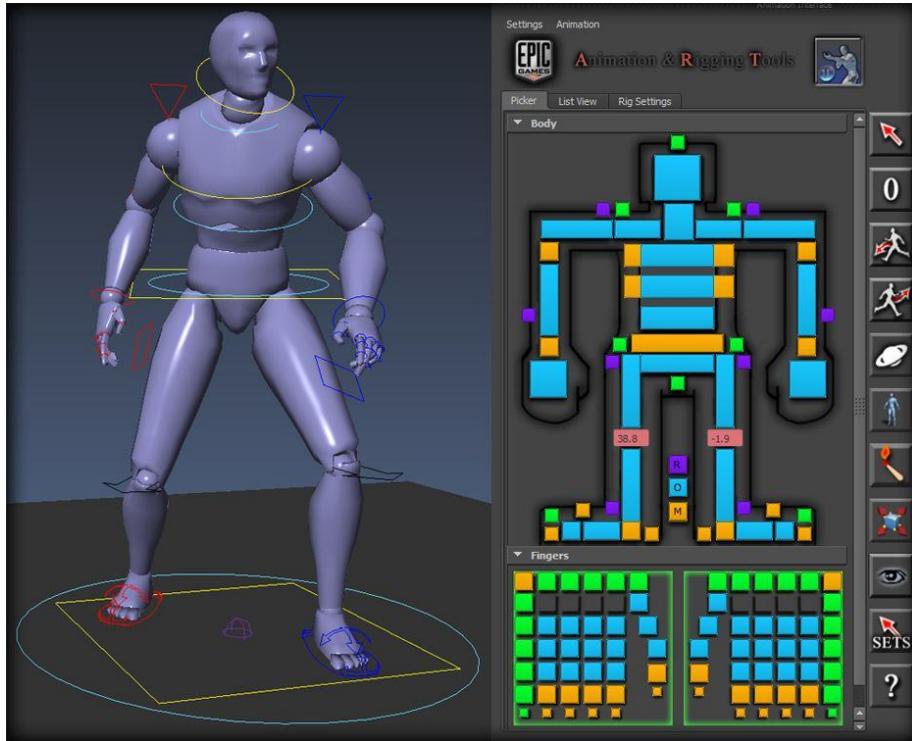
# Required external tool: Maya

Other programs like 3DS MAX and Blender, but rigging is only supported in Maya.  
The thing is, that it only works on *normal* Maya and not the **LT** version, because it  
needs python support.

The tool is called “ART”.



# Animation and Rigging Tools



# Some things that have been done by me

- Articles on the wiki
- Various answers and discussion participation on answerhub and forums
- Contributions
- Tutorials + Guides

PR #2470

Android local notification support



And

**Unreal Engine 4**

**Ratchet and Clank  
Tutorial Series**

**Weapon System**

(You can find these and more on my  
YouTube channel)

# Greatest bill of learning resources

<https://unreal-engine-4.zeef.com/tom.looman>

**ZEEF**  
Find information through people

Search... Login Create a page

Unreal Engine 4 by  Tom Looman

★ Follow - 282

 unreal programming engines cpp games

About this page Table of contents Suggest 204 Share Unreal Engine 4 f t g+ ...

**Introduction**  
This page contains a collection of tutorials, articles, downloads, blogs, tips, and a wealth of other Unreal Engine 4 resources that I have stumbled across and would like to share with the Unreal community.  
You can suggest additional links at the bottom of each block.

**UE4 Plugins**  


**Learning Unreal**

- 1 Getting Started Guide
- 2 YouTube (Official Channel) 
- 3 Tesla Dev
- 4 MetalGameStudios
- 5 Crash Course
- 6 Wiki
- 7 Forums
- 8 Getting started with the API

more + <>

**Learning Blueprint**  


- 1 Introduction to Blueprints
- 2 Romero
- 3 Blueprint Macros
- 4 Managing complexity
- 5 Editor Tips and Tricks
- 6 API Reference
- 7 Learning Game Templates
- 8 Shooter Game Tutorial

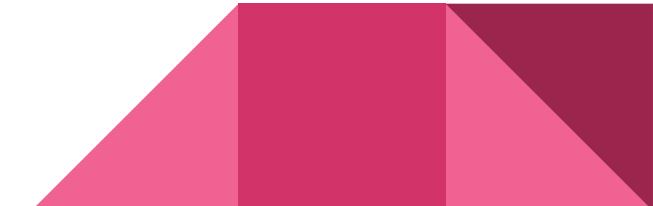
+ <>

# But in reality

Everything is much smoother as soon as you get a good understanding first.

Don't just copy paste stuff over the internet or follow instructions blindly!

Community is much helpful on Discord, AnswerHub or forums.



# Creating your first project

Live Demo :)



# UNREAL EDITOR

Unreal Editor 4.14.3

Copyright © 1998-2016 Epic Games, Inc. All rights reserved.



Initializing... 0%

# My experience

You are going to hate it at first.

You need to make a complete small game up to production in order to be proficient enough (~7 months) to do work at a decent pace, without having to wrestle with the engine

TAKE THE ENGINE WITH A GRAIN OF SALT! IT'S NOT FOR EVERYBODY!

# Let's say I know somebody that...

Just finished university with a thesis on Game Development with UE4. Within a month he got job offers from:

- Nickelodeon
  - Zynga
  - ArenaNet
  - Pixar
  - Dreamworks
  - Wargaming
  - Technicolor
  - Activision
- and much more small(er) VR startups

# ¿QUESTIONS?



# Thank You!

